

DRAGON WORLD

No. 2 February 84 60p



SPECIAL FEATURES
Software
Reviewed

INTRODUCTION TO 'DRAGON WORLD'



Welcome Dragon users, to our magazine especially if you are one of the thousands of new users to write our last publication in December.

1984 will be an interesting year, and starts with the news of a major order we have just secured to supply a customer in Spain. With that in mind, it would seem a particularly good time to extend a warm welcome to all our new overseas readers and to say how much we look forward to hearing from you and to receiving any contributions you may have for Dragon World.

This month we are also featuring "AMPALSOFT", an educational software company responsible for several interesting titles which we highly recommend to Dragon users.

I do hope you are enjoying our new magazine and if you have any ideas for improvement, please let us know.

May I wish you every success in 1984 and many happy hours with your Dragon computer.

SUBSCRIPTION OFFER

SUBSCRIPTIONS FOR DRAGON WORLD CAN BE OBTAINED DIRECT FROM DRAGON DATA FOR £4. THIS WILL ENTITLE YOU TO 6 BI-MONTHLY ISSUES MAILED DIRECTLY TO YOU. AND THE FORM FOR THIS CAN BE FOUND IN A SEPARATE INSERT TO THIS MAGAZINE. IF YOU KNOW OF ANYBODY WHO MIGHT BE INTERESTED WHO CURRENTLY DOESN'T RECEIVE THE MAGAZINE, PLEASE LET THEM KNOW OF THIS FACILITY.

CONTENTS

Managing Director's Column	3
Editorial	3
User Club News	6
Book Drive Presentation	7
An Introduction to Ampalsoft	9
Dragon Answers Back	11
Machine Code Corner	14
Young Users' Pages	16
Dragon 84 Review	19
This Month	26

EDITORIAL

Welcome to the first issue of "Dragon World" for 1984. We would like to take this opportunity to wish all our readers a happy and prosperous New Year, and to welcome all the new users to this, the first issue of "Dragon World". Thank you to all the users who have sent in letters and contributions which were, and still are, gratefully received.

In this month's issue you will find a complete software list. If you are having difficulty obtaining software from the dealers and wish to buy direct, please post to: Mail Order P.O. Box 40, Port Talbot SA13 123 West Glamorgan.

We will accept cheques, Access, American Express or Diners Card numbers. Orders may be taken over the telephone on 0656 744700, ext. 225 quoting relevant card number. We regret that no Barclaycard numbers can be accepted at present. The poster offer is still available, and we will continue to include both these offers in future issues of the magazine.

We should like to thank all the users who entered our "Dragon User of 1983" competition. There were many entries and it will obviously take time to decide on a winner from the many varied applications. We are at present compiling a short list and the results of the competition should be announced in the next issue of "Dragon World" in April.

In order that we can keep our mailing list up-to-date, please inform us of any change of address or machine replacement. For those readers who have sold their Dragon, or intend to do so, please inform us so that we can discontinue your "Dragon World". If you do sell your Dragon, the new owner may wish to receive this magazine, so do please inform them of our subscription offer so that they do not lose out.

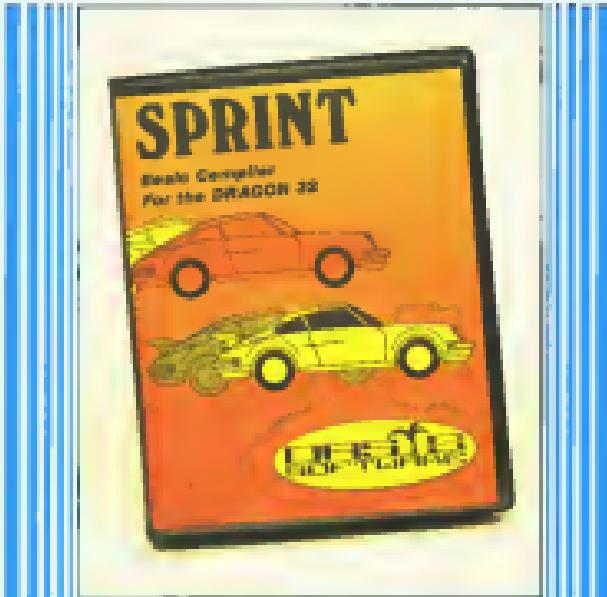
Editorial Team,
Dragon World,
Dragon Data Ltd.,
Kemp Industrial Estate,
Margetts,
Port Talbot, SA13 2PE,
West Glam
Tel. 0656 744700





Oasis Software present...

The first basic compiler for the DRAGON 32



SPRINT BASIC COMPILER

by Dr David Gray PGCert (Edin) DRAGON 32

Dragon Basic is a standard version of DragonBASIC which

conveniently converts standard BASIC into virtually everything

except floating point arithmetic and associated commands. All

variables, integers and the Dragon sound and graphics

commands are fully supported.

The fast-released version of the approach uses UCB 32D Pascal where the basic program is in relocatable intermediate

code and then retranslated using a run-time package which is

switched off the start of the compiled program.

- Programs written in L16 basic code
- Almost the entire basic is supported with the exception of floating point commands.
- Code produced will run independently of the compiler (for potential portability)
- Programs uncomplicated from the user remote control and the much longer program can be compiled
- SPRINT is a vector-based system and a component module command is included
- Free demonstration program with each program bought to illustrate the full power of the Compiler
- All OASIS products are covered by a written guarantee

JUST RELEASED

DRAGON BASIC 32

Dragon Basic is an enhanced version of the standard programming language

which adds a lot of its many features including...

- A complete set of structured programming constructs
- IF...THEN...ELSE...END IF CASE...OF
- COMPUTER, EDITOR and SOURCE simultaneously makes the most of the Dragon's display and ease of use
- Very rapid compilation (Sprint can actually compile code nearly as fast as DR32 BASIC)
- Fully recursive
- Standard libraries with support for graphics including routines which demonstrate techniques for manipulating pixel and graphics word functions such as PSET and CIRCLE

OASIS SOFTWARE, Alexandra Parade,
Merton, Surrey, SM4 5DT, UK. 0161 467 1071

Please send me

SPRINT Basic Compiler £14.95

DRAGON BASIC 32 £14.95

Name

Address

After 10.00 orders to
taken by phone
24 hours a day — 0161 467 1071

DRAGON 32



DRAGON CHESS £9.95



- Six levels of play
- All legal chess moves including en passant, castling and pawn to queen promotion
- List of previous moves stored which can be displayed or printed
- Loading and saving of games positions from tape
- High resolution graphics which can be flipped round to match black or white play from either end
- Simultaneous text and machine

Propositions in which the last value from the marked lines are included

Brugge beschikt over een
samenhangende historische kern.

- States can be exchanged at any stage
- Best move first
- Set up from any position
- Opening move theory
- Moves may be taken back and play resumed from any point
- Change level of play at any point in the game
- Will adjudicate (parties between humans)
- Very high standard of play
- Professional packaging and little movement

"I thought this was an excellent question
at the game. What was happening?"

BACKGAMMON REGS

OTHERS
15/16

INVADER CUE 600

DOMINOES



The next lesson goes at
the beginning principle with the
most difficult concept.

At least, Orlitzky and Kishimoto et al. (2007) argue that the environmental costs of organic food are not necessarily higher than those of conventional food.

As well as being one of the last groups of institutions to take to the Internet, the industry seems to be the last group to move on to e-commerce.

Two groups for one with the
transformation. It is not an independent
but their programs have different
titles and also the original however
comes to a point to be the second of your choice

Mind Games Compendium —
All five games for just £19.95

ALL DAIRY PRODUCTS INCLUDE
RAT AND FLEA BAIT TRAPS
DAIRY PRODUCT IS PEST-PROOFED
DEEP-ITCHING AND REAVERS

If your local dealer does not stock these products then let us know his address and we will contact him.

**ACCESS ORDERS
TAKEN BY PHONE
24 HOURS A DAY**



1000.000000

—
Bacon, John (1619-1691)
English pamphleteer, historian
and author. Known as
"The Devil's Advocate".
His most famous work

Please send me
DRAGON CHESS
BACKGAMMON
OTHELLO
DOMINOES
INVADE CURE
MIND GAMES
COMPUTER GAMES

Enclose cheque/P.O. for £
NAME _____
ADDRESS _____

OASIS SOFTWARE

Intelligence in code.



USER CLUB NEWS

Firstly I should like to welcome all new users to the User Club pages. The object of these pages is to give details of users' clubs and groups as already stated in previous issues.

Secondly I should like to apologise to Mr J. W. Smith of West Yorks whose details I included in the last issue of 'Dragon World'. Mr. Smith no longer owns a Dragon 32 and does not wish any further contact from Dragon users. I should also like to apologise to all the people who contacted Mr. Smith who will be forwarding your letters to me and I shall try to answer each one personally. In future extreme care will be taken so that a similar misunderstanding does not occur.

Can I also include here a plea — if you sell your Dragon, move address, or have any other altered information from your registration card, please let me know so that I can adjust the mailing list accordingly.

The response to my request for information on clubs not already included on my list has been slow but with Christmas and the New Year I am not too surprised. If you have written to me and are not included, please take into consideration the fact that this column is written over a month before you receive Dragon World.

The new clubs I have heard from are:

SOUTH WALES

Anyone interested in forming a Dragon users club in the Llandaff area, please contact Mr. W. R. Collins on Llone R. (056421 50917) (home).

ENGLAND

Cornwall

Mr. P. M. Starke
41 Old Roselyn Road,
Middlewear,
Plym-

Devon PL24 3LN
Tel. Ply 4923

Mr. Starke is interested in hearing from anyone in the area as he is proposing starting a Dragon Users Club.

Cheshire

Elesmere Port Computer Club
Chairman: Mr. G. Corcoran

Tel. 051 327 3912
Meets: alternate Mondays, 7.00pm, at Chelch Hall
adjacent to Elesmere Port Golf Club

Lancs

OCUS (Oldham Computers Users Society),
Meetings:
Caldwell Community Centre,
Rochdale Road,
Oldham
every Tuesday 7.30-9.30pm

West Midlands

John Lucas (0809 West Midls)
Tel. Sedgley 72521

This is a small club that does not really wish to expand a great deal, but they are prepared to act as a 'springboard' for other users in the area to get together in a small geographical unit.

NORTHERN IRELAND

The last issue gave Mr. Peter Leech's details and how he has requested that his telephone number is included. You can contact Mr. Leech on Antrim 88045.

On the Air

There is a club, possibly to be called 'Dragon', now operating on Amateur Radio Frequency allocations. The club had its first try out in early January and immediately gained 10 members. The 'net' is held on radio frequency 144.525 MHz each Sunday morning at 11.30am (soon to be altered to 11.00am). The net is officially called 'computer users net' and has been set up by an enthusiastic Dragon owner. I am sure that this club will be of interest to the many radio amateur operators who are also Dragon owners.

Christmas Fair

It was good to meet Doug and Dave from the Dragon Independent Owners Association at the Christmas Fair, Wembley. It is always helpful to get exchanges of ideas and hopefully some of the suggestions discussed may be implemented in the near future.

Even if clubs are unable to visit the exhibitions we are present at, there are still other methods of communication and I should appreciate ideas and suggestions from the clubs as to what they would like to see on these pages and within the magazine.

Thank you to all those club members who gave all some of their Saturday to help us out by demonstrating in larger stores. Please make sure that you send your reports in though!

Cathy Hyde

Cathy Hyde



DRAGON CHRISTMAS DRAW PRIZE WINNERS

1st PRIZE: Mr. M. G. Myett, Gelli, Rhondda



Pictured above is Kevin Stephens, a marketing executive at Dragon Data, presenting the first prize of a double disk drive to the lucky Dragon 32 owner, Mr. M. G. Myett (extreme right). Also pictured are Mr. Myett's wife, Carol, and his two children, Christopher (aged 9) and Teresa (aged 11).

The Editorial Team at Dragon World would like to congratulate the Myett family on their success and hope it enabled them to enjoy the festive season even more than they usually do.

In addition, Dragon Data offered 20 runners-up prizes of £20 free software to be chosen from our catalogue. The 20 lucky Dragon users pulled out of the hat were as follows:

Mr P Francis, Pontypool	Mr J Griffin
Mr Valentine, Congleton	Hull
Mr S C Lowe, Lerbens	Mr J Wall,
Mr E Perry, Atherstone	Dyfed
Mr J A Bulwer Walsall	Mr A L Hemsley
Mr J Bull, Marsfield	Rugby
Mr D Foster, Leicester	Mr A P Jennings
Mr B Wallis Hull	Harrow
Mr G Hesson Lands	Mr D M Leary
Mr J Tupper, Gantham	Worcester
	Mr W Meliss
	Darwen
	Mr G F Spragg
	Middleboro
	Mr J F Greenwood,
	Bolmont
	Mr P A Dow
	Leeds
	Mr Van Loveren
	Medessem, Belgium

Congratulations to all of you

DRAGON PUZZLE 4

Here is the solution to the Christmas puzzle complete with the revealing hint!

```
10CLS.PRTNT@10, "DRAGON PUZZLE 4"
20PRINT@250, "FRANKINCENSE"
30PRINT@320, "GOLD"
40PRINT@160, "CAKE"
50PRINT@380, "SNOW"
60PRINT@190, "STAR"
70PRINT@90, "MESSAGE"
80PRINT@220, "POST"
90PRINT@291, "PRESENT"
100PRINT@100, "PEARTREE"
110FORI=3 TO 11 FOR J=6 TO 8 STEP 2
120X=32*I+J P=PEEK(X+1234)
PRINT@X,CHR$P+321, NEXTJ,I
130PRINT@400,""
```

Errors — Dragon World issue 1

There were, unfortunately some typesetting errors in the first issue of Dragon World. Whilst we continue to investigate alternative ways of presenting programs which are both error-free and readable, perhaps you will accept our apologies together with the following list of errors.

Fortunately, most of the errors were easy to rectify given some experience but we understand how frustrating it can be for newcomers to be confronted with '5N ERROR' etc. after painstaking typing sessions!

It may be helpful to review some of the common problems due to typesetting. Spaces often are a problem as they are not as wide as a character. Try to leave spaces between variable names and BASIC command words such as 'TO' as in 'FOR I=1 TO N'. Also note that the space bar is often used as a control under INKEY\$ and that it is important to distinguish between quotes around nothing and quotes around a space.

Here then is a list of the mistakes:

TORNADO

```
Line 180 — a space between UI and THEN.
Line 200 IF F<8 then F=8
Line 360 — the second semi-colon should be a colon
Line 380 — A final quote is required
Line 390 — The last two pairs of quotes should surround a space..
```

CHRISTMAS TREE

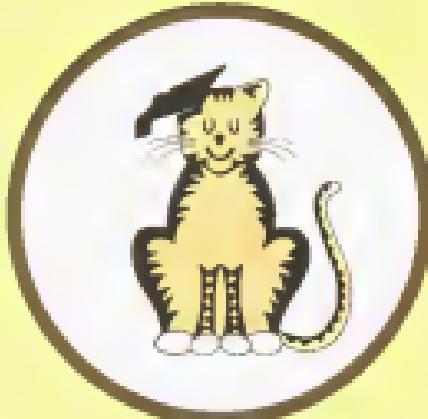
Line 390 — the second comma in PAINT(X,Y) should not be present.

HOLES line 1020 should start with PRINT@480,MIDS(N\$(),1+2*N(),).

CHESHIRE CAT EDUCATIONAL SERIES

from

AMPALSOFT



CHESHIRE CAT

The First name in Educational Software.

An exciting range of top quality programs
covering all needs from pre-school to 'A' level.

Now available from Boots and other good stores.

Amplsoft Computer Services Ltd.
31 Woodbridge Road, Derby Green, Bingley,
West Yorkshire, BD11 1JL
Tel: (0223) 876677

AN INTRODUCTION TO AMPALSOFT



An exciting and rapidly expanding range of educational software for the Dragon 32 home computer is being produced by a novel co-operative of housewives and teachers.

This unlikely combination of talents has already shaken the software market with their high-quality polished programs of genuine educational benefit, presented in the finest moulded plastic book-shaped packaging under a distinctive Cheshire Cat logo.

The motor-board-sporting cat featured on the cover already commands pride of place in Boots' stores, stocking computer software and other national retail chains and specialist software stores are showing a keen interest.

Yet 15 months ago Cheshire Cat was just the brainwave of three mothers with an active and enquiring collective brood of seven young children. Each family, all close friends, had bought Dragon 32 home computers because they realised the impact new technology would be having on their offspring's education and future employment prospects.

Naturally anxious to provide an early opportunity for their children to acquire some of the skills the microchip revolution would require, they had sought the machine with the best keyboard for young fingers to operate. At the same time they also intended to combine that with the largest memory, best colour graphics and sound facilities available. At the price the Dragon 32 topped the poll.

But very soon the trio of housewives — Patricia Lansdowne, Ann Mortimer and Lynn Nixon,

discovered there was a severe limit to the number of worthwhile educational programs readily available. Software shelves groaned under an avalanche of arcade-type games but genuine educational programs were rare. So Lynn, a skilled programmer together with her two chums, had the bright idea of roping in a teacher friend to devise a bright, colourful and animated maths program suitable for their own four to six-year-olds.

Maths 1 went so good, neighbours and friends started clamouring for copies and Cheshire Cat was born.

Then a brilliant but simple Basic Tutorial program to teach beginners how to program their own Dragon 32 was produced — and that hit now sold over 20,000 copies. "Basic Tutorial" has also featured in two major national advertising and promotional campaigns by Boots as part of a software pack sold with each Dragon 32.

Realising the tremendous potential this barren market place presented, the girls formed Ampalsoft at Knutsford, Cheshire, in October 1982, and devised their own Cheshire Cat logo and Ampalsoft trademark.

The talents of their husbands were swiftly drafted along with teachers and programming friends into the expanding co-operative fold. And it was the husbands who provided the sound business base on which Cheshire Cat's commendable success is firmly founded — Chris Lansdowne, 35, is a financial director, Colin Mortimer, 35, an electronics design engineer with a special interest in computers, and Tim Nixon is a top sales consultant. Now this multi-talented team includes more than 40 teachers and programmers working on new educational products for the Dragon 32 to add to the nine strong present range.

Cheshire Cat director, Patricia Lansdowne, credits their meteoric rise, with £1 million sales in their first year, to producing such top-class products for a virgin market.

Patricia said: "We started Cheshire Cat to provide our own children with the educational software we could not buy. We have never cut corners on quality or detail and we never will. Each program takes up to four teachers and three programmers three months to complete and test. Our driving force is our children's future, not huge profits."

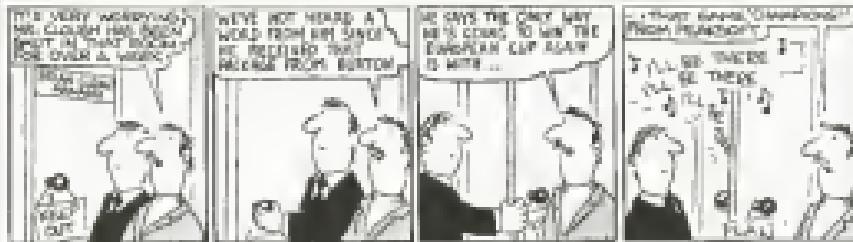
The ambitious target Cheshire Cat have set themselves is to eventually produce a comprehensive range of programs to provide every school subject from pre-school age ability right through to 'A' Level standard.

A wholly feasible program explores the 22 major areas of the current maths syllabuses of all the examination boards for 'O' level, CSE and Over 16 examinations.

Also available is the thrilling history simulation for 11-14 year olds, "Superspy", a first school reading aid, "Early Reading", the top selling

(continued on page 28)

BEHIND THE SCENES AT PEAKSOFT



RESEARCH BY EAST 18-19 TEAMS THIS BILLY. AVERAGE DELAY 2000. PREMIERES IN THREE BOWLS.

osse

PROBLEMS



PEAKSOFT FREIGHT Version Two 0111-000

10

Yester day I was surprised, and it is a
long story how. Can you believe the
injustice, the ignorance, the
and the general stupidit. Why don't
and then with the legal language.

26.85

THE BOSTONIAN

Other religious and ceremonial sites
3. Residential Groups
Human and Animal
Burials

Two-dimensional patterns

1000000000

References

WORCESTER STATE — Unique low cost state leadership education as you concentrate your study pursued by the nation's finest. Statewide is a statewide graduate education as you learn in the Mass. Law to ensure your Commonwealth's future leaders.

What is the relationship of the two? — The classical errors are associated with the extremes and poor quality that you can expect to observe. The types of the errors are largely based on the nature of a Manager's behavior and you need a manager of theory to convert a theory to implement theory. If the manager is not interested

Source: T. P. H. Report 2004-05 — Fewer smoking admissions at one respiratory centre. Melbourne: Department of Health and Human Services, Victorian Health Promotions Agency.

10

DRAGON ANSWERS BACK

Question

I wish to use the motor control relay in my Dragon 32 by means of the cassette lead to perform switching operations on external devices. I find that using the 'MOTOR ON' and 'MOTOR OFF' commands in basic give only a terribly slow operation of the relay. You give the addresses for these operations as &H8015 and &H818 and I have tried POKEs of all numbers from 01 to 255 into these, but nothing happens. Is it possible to speed up the operation of the relay, please, preferably to the speed of light, or near, with some simple POKEs?

Answer

To speed up the operation of the cassette motor relay, you have to access the locations H8X95 and 96 which control the delay. If you poke &H95, you will get 140, whereas &H96 gives 150.

If you POKE &H95 with 01 and &H96 with 01, it gives the optimum speed available.

The locations &H8015 and &H8018 just control the on/off routines and do not need to be altered.

Question

I have recently purchased a Dragon 32 and am very pleased with the operation of the BASIC and the quality of the software.

My major grievance is that you cannot, so I am lead to believe, mix graphics and text on the high-res screen.

Answer

It is possible to mix text and graphics in an alpha semi-graphics mode, PMODE 24, which gives a resolution of 192 x 64. Details can be obtained on request (please send self).

Alternatively, you could type

POKE &HFFC0,0

POKE &HFFC3,0

POKE &HFFC5,0

This will enable you to put text characters on the graphics screen starting at &H4000, by poking the respective character codes at the desired locations.

Question

I recently purchased an Epson RS-80 printer to work in conjunction with my Dragon 32. Whilst the printer works perfectly using the commands

LIST

7H->2,1,..,

7H->2,0HSH1,..,

I am puzzled to find that there is no information in the printer manual on graphic screen dumps. Could you possibly give me any assistance in this matter?

Answer

We have available, free of charge, screen dumps for the Epson MX-100 and also for the two popular ST/KO251A printers, the GP-100 and GP-200, one of which is included in this issue. The MX-100 dump should be easily adaptable for most Epson machines. These routines are available on request (please send SAE).

Question

I am a radio amateur and am writing to ask if you market, or are aware of, a program which would allow my Dragon 32 to send and receive RTTY signals, hopefully on cartridge.

Answer

We are aware of a high quality RTTY program on cartridge and cassette — priced at £12.00 (cartridge) or £21.00 (cartridge).

The program is written wholly in Machine Code, and occupies approximately 4K of memory. The audio signals, up to over 160 baud, are fed directly into the cassette input line. Alternatively an external terminal interface can be used for higher speed.

The program also includes a 4096 character "type-ahead" buffer which allows you to reply while receiving.

For full details apply to:
Mr M. J. Kenny,
22 Grosvenor Road,
Seaford,
East Sussex
Tel: 0323 883078.

INSTRUCTION COURSES

Dragon Data are considering offering the following courses:

2-Day DEALER SALES COURSES

Cost: £280+VAT price includes one night's accommodation and meals.

Aim: To help dealers maximise sales and service.

1-Day INTRODUCTION TO THE DRAGON

Cost: £25

Aim: To enable new users to make the best use of their Dragon system and to become more conversant with BASIC.

Venues to be decided.

If you are interested in receiving further details please write to Kathy Ashton at Dragon Data.



MACHINE CODE CORNER

This issue we consider the problem of transforming the contents of a video screen to a matrix dot printer. Most such printers these days can operate in 'graphics mode' whereby each byte of information sent to the printer is interpreted as a pattern of seven or eight vertical dots. A graphics screen can then be reproduced on paper by transforming the information content pixel by pixel into successive bytes, each byte to communicate to the printer a particular pattern of vertical dots.

Unfortunately, different printers require the information to be coded in different ways and so it is impossible to provide a single universal program for all printers. In this article we consider two popular printers — the Sankyo CP1200A (referred to as SC) and the Shinwa CP80 (denoted by CP).

First let's tackle the problem from BASIC where communication to the printer is via the use of PRINTER# 2. A unit of information is one byte and this can be represented by any decimal number from 0 to 255. Thus PRINTER#2:CHR\$N sends to the printer the byte whose binary form equals the decimal number N.

Some of the values of N are used by the printer as CONTROL CODES, and in particular each printer uses a particular set of such codes to change to graphics mode. For SC, N=8 instructs the printer to accept all subsequent bytes as graphics bytes until the control code N=16 is sent. For CP, the control codes are more involved and are N=27, followed by 48 (corresponding to '1') to control the line feed width, then N=27, followed by 187 ("K") plus two numbers L and M to instruct the printer to interpret the following L+256*M bytes as graphics bytes.

The differences between SC and CP do not stop here! SC uses a 7 bit column form and such a column escape first, third, fourth, sixth and seventh bits are 'set' would require the code 128 (always present) + 1 + 4 + 8 + 32 + 64 = 237. In other words as we descend the column of dots we add to 128 a power of two for every dot to be printed using the powers 1,2,4,8,16,32,64 with 1 associated with the top bit and 64 with the bottom.

On the other hand CP uses an eight bit column with the top bit associated with 128, and then down through 64,32,16,8,4,2 and 1 for the bottom bit.

First then we construct a program for SC assuming that our graphics picture has been

created in PMODE4 with black on a buff background. The pixel at X,Y is set to black if PPOINT(X,Y)=0. We construct a program that scans down each column of seven pixels starting with Y co-ordinate J=0 increasing to 189 in steps of 7, and X co-ordinate I=0 to 255, the column being scanned by variable K going from 0 to L (where L is normally 6 except for the last line). The summation of the N value corresponding to each column configuration is done in the K loop using P= P+PPOINT(I,J+K)=2^I*PK. Note that we have taken the trouble to place the powers of 2 in an array — this is not just for neat programming but an important necessity when considerations of speed are relevant (see the article on DATA statements in the first issue of DRAGON WORLD). The subroutine is as follows:

```

1 TE=189:FOR I=1 TO 7:READ P#NEXTI
2 DATA 1,2,4,8,16,32,64
10 PMODE4,1:SCREEN1,1:PCLS1-COLOR8,1
20 LINE(0,0)-(255,191),PSET,B
100 PRINT#-2,CHR$81:L=8
FOR J=0 TO 189 STEP 7
200 IF J=189 THEN L=2
300 FOR I=0 TO 255:P=TE:FOR K=0 TO L
400 P=P+PPOINT(I,J+K)=2^I*PK:NEXTK
500 PRINT#-2,CHR$P,
NEXTI:PRINT#-2:NEXTJ
600 PRINT#-2,CHR$1151

```

Since CP uses an 8 bit graphics code a different program can be written for CP using the graphics bytes as stored inside Dragon providing we agree to represent the screen on paper sideways. To be precise, the first row of the printed picture will correspond to the left-most vertical strip of the screen. If we again use PMODE4, but this time printing BUFF as BLACK then each byte of the graphics memory corresponds exactly to the information returned by CP. The program uses Dragon's own store of the address of the start of the current graphics screen (bytes Hex 0A and 0B).

```

10 PMODE4,1:SCREEN1,1:PCLS
20 LINE(0,0)-(255,191),PSET,B

```

```

100 ST=256:PEEK(0H+H8A)+PEEK(0H+H8B)
110 PRINT#-2,CHR$271,"1"
120 FOR I=0 TO 21:PRINT#-2,CHR$271;"K",
CHR$182,CHR$182,
130 FOR J=191 TO 0 STEP -1
PRINT#-2,CHR$(PEEK(ST+32+J+I));
-NEXTJ
140 PRINT#-2,CHR$(13),NEXTI
150 PRINT#-2,CHR$(27),"?"

```

And so at long last to Machine code! We look in detail at the construction of a machine code program for CP and present a BASIC program to PMODE4 in the equivalent program for SC. Since the speed of execution of these programs is now

determined by the speed of the printer itself and not by the program design, both construct the screen dump horizontally and assume a block on buffer format in PMODE4.

We need to be able to send bytes of information to the printer from machine code — this is done by loading A with the appropriate byte and then the instruction JSR \$8000F uses a subroutine in ROM that sends the byte in A to the printer. We use this procedure for both control codes and graphics bytes.

The basic architecture of the program is constructed by considering those eight bytes in RAM corresponding to a current rectangle of the screen consisting of 8 by 8 pixels. These eight bytes will occur in RAM at the positions *start* + 32, *start* + 64 and so on. To assemble a byte of information the first bit of all the eight bytes must be interrogated, then the second bit, and so on. When all eight bits have been assembled we move on to the next set of 8 bytes corresponding to 8 vertical pixels adjacent to the previous 8, and after 32 such sets we will have assembled 8 * 32 = 256 bytes of information corresponding to a complete horizontal strip of the screen 8 pixels deep. These 256 bytes are stored in RAM from hex 21F00 to 21FFF.

10	ORG	\$10000
20	PUT	\$10000
30	ST1	SD120
40	ST2	SD124
50	LDA	#\$10
60	JSR	\$8000F
70	LDA	#\$10
80	JSR	\$8000F
90	LDD	SD1A
100	STD	SD1A
110	TFR	D,X
120	LOOP1	LDY #\$1F00 100E7F00
130	LOOP1	TFR X,0
140	ADC0	#\$20
150	STD	ST1
160	LDA	#\$80
170	LOOP2	CLRB
180	TFR	X,U
190	LOOP3	LSLB
200	BITA	.U
210	BNE	CONT
220	INC0	SD
230	CONT	LEAU
240	CMPIU	ST1 11037E20
250	BUD	LOOP3
260	STB	.Y+
270	CMFY	#\$80000 100C10000
280	BD0	CHECK
290	CMFA	#\$01
300	BD0	NEXT
310	LSRA	44
320	BRA	LOOP2
330	NEXT	LEAU 1,X
340	BRA	LOOP1
350	CHECK	LDY #\$1F00 100E7F00
360	LOOPC	LDA .Y+

370	BNE	FOUND	10409
380	CMFY	#\$80000 100C10000	100C10000
390	BLO	LOOPC	25F6
400	BRA	INC	2021
410	FOUND	LDY #\$1F00 100E7F00	100E7F00
420	LDA	#\$10	861B
430	JSR	\$8000F	SD600F
440	LDA	#\$40	861B
450	JSR	\$8000F	SD600F
460	CLRA		4F
470	JSR	\$8000F	SD600F
480	INCA		4C
490	JSR	\$8000F	SD600F
500	LOOPF	LDA .Y+	AS43
510	JSR	\$8000F	SD600F
520	CMFY	#\$80000 100C10000	100C10000
530	BLO	LOOPF	25F6
540	INC	LDA #\$00	8600
550	JSR	\$8000F	SD600F
560	LDD	ST2	FC7E4
570	INCA		4C
580	CMFD	#\$7	100007
590	BLD	LOOP	1005F807
600	LDA	#\$10	861B
610	JSR	\$8000F	SD600F
620	LDA	#\$02	8622
630	JSR	\$8000F	SD600F
640	RTS		39

The finer details are as follows: ST2 is used to store the address of the first byte of a strip, and is initially supplied with the address from store S8A (lines 900, 1000). This is incremented by 256 before returning to 'LOOP' by loading D from ST2 and incrementing A (as A is the most significant byte of D). Given the address X of a current set of 8 bytes U is used to control looping through the 8 bytes (change line 230 increments U by 32) and ST1 is used to indicate when this procedure should stop by setting ST1 to X + 225 (lines 130-150).

To access the individual bits of our current set of bytes, we use A to provide a 'mask' as that for example to examine the most significant (high-mosdi) bit A is set to have #1 (binary 1000000000000001) (line 140) and subsequently changed to the other masks using LSRA (line 210). LOOP2 controls this process with branching to NEXT when A = \$21 (mask 0000000000011).

To assemble the graphics information B is first set to zero. The command BITA .U (line 280) checks to see whether the appropriate bit (determined by A) of the current byte (determined by U) is set — if so 1 is added to B. On re-entry to LOOP3, LSLB shifts the information one place to the left so that the first bit (highest on the screen) eventually ends up as the most significant bit. When a byte of information is complete it is stored in address Y and Y is then incremented. A check for the end of a strip is then possible by comparing Y with \$80000.

Note finally that lines 360-390 provide a check to see whether the current strip is blank or not — if so all that is required is a line feed, otherwise the whole strip is printed by loading each byte from



YOUNG USER'S PAGES

MOVING STRINGS

Welcome to all our new readers. Dragons will have appeared in many homes over Christmas and many of you will be programming for the first time. On these pages we offer you short programs which help you explore the possibilities of your Dragon. We usually concentrate on one or two commands and this week, for the new readers, we look at the PRINT#1 command but then we will use it to make pictures that move. All the commands used are explained in detail in the programming book which comes with your Dragon. Before you make something move you must be able to show it on the screen. I've chosen to use the PRINT screen. What can you print on this screen? Well if you look at pages 136-138 in the prog ramming book you will see a list of characters and their code numbers. There are letters, numbers, punctuation marks and some special shapes which can be printed in different colours. If you haven't experimented with these yet just try!

PRINT CHR\$(95)

Then press <ENTER>. A letter A should appear below the command line. If you want to print it in a different place you use the PRINT#1 command. Type

PRINT#1 272,CHR\$(95)

When you enter this the A will appear in the middle of the screen at position 272. There are 512 positions numbered from 0 to 511. You can refer to page 148 of the programming book to find the positions on the grid there. The characters are numbered from 0 to 255. Some of the codes don't actually print anything — some give instructions for printing like 'go to the next line' or 'go back a space'. The characters which have codes above 127 are block shapes on a coloured ground. If you add 16 to a code you get the same shape with a different background. I'm going to use these characters to make a train.

First the carriage. CHR\$(129) is shaped like this █ and CHR\$(131) like this █. We will write them as C-BS(129+N) and CH-BS(131+N) then we can change the value of N until we have a colour we like. We can put the characters together to form a STRING and give the STRING a name. A string is just some characters put together and a name allows us to refer to them without writing the whole string. A string variable must have a name which ends with \$. I have called the \$C for carriage. Try this tiny program.

100 CLSS

500 CS-CHR\$(129+N)+CHR\$(131+N)

1000 PRINT#1 CS,CS

GO TO 100

There is CS. But the program stops and you get OK on the screen. To prevent this type in a line which sends the program round in circles until you press the <BREAK> key.

110 GO TO 110

N=32 GO TO 100

Line 110 goes round in circles and the next line gives N a value and starts the program off. See the colour change? Use <BREAK> to stop the program.

Three Cs in a row look like three windows but we need a CHR\$(130+N) to finish it off and something for a coupling. I chose an equals sign but you may prefer an arch. Now type these lines

500 CS-CS-CS+CHR\$(130+N)+"-"

N=112 GO TO 100

There — a carriage top with orange windows.

As you can see if the character is a keyboard symbol like "equals" we can add it to the string in quotes. In fact CHR\$(81) and "=" are equivalent.

What about the base of the carriage? How can we make the wheels? Well there are quite a few round things on the keyboard (B D " and @). I chose @ and I've filled in the gaps with = so that they can't get lost in printing! (But you'll see later that they have another purpose.) We can make a string \$S for the base using the quote marks and we can print it under the carriage — that means starting at a position 32 more. Add these lines

700 CS="@=@@=@=@@@"

110 PRINT#1 32,\$S

N=34 GO TO 100

To get the carriage moving we should have an engine, but I'll come to that later. For now let's get that carriage to move across the screen. To do this we must print it in different positions. If we call the print position for the carriage top P and the position for the base P+32 we can after P in a loop. We will overwrite lines 100 and 110. At the same time we can sort out another problem. If you RUN your program now you will see that the whole line to the right of the carriage turns green. If we finish the PRINT command with a semi-colon this will stop. Here are the new lines

100 FOR P=180 TO 120

110 PRINT#1 P,CS; PRINT#1 P+32,\$S;

120 NEXT

RUN

Well it moved but grew as well because we printed each carriage on top of the last without erasing it out. We can get over this by putting a blank character behind the carriage when we PRINT so that it rubs out as it goes along. The new line 110 is

110 PRINT@P, " "+CS: PRINT@P+22,
" "+BS.

Make sure you have a blank space inside the quotes. Now RUN. Success at last!

What about the eraser? Well I'm sure you can design one for yourself but here are the lines for mine:

100 ES=CHR\$1(128)+STRINGS(5,128)
+CHR\$(123+N)
102 BS\$="000000000000"

There I introduced a new command, STRINGS(5,128). This made a STRING of five black rectangles — the first number is the length of the STRING and the second number is the character code. (That second number could be a one character variable or the character itself enclosed in quotes.)

Now you've seen how to make a STRING move you can put several commands and the engine into one STRING and make them all move.

I decided to make my train go right across the screen and then come back on again as though it was going round and round on a track. So first I made a string of four carriages and an engine. Then I added a lot of blanks, CHR\$(140) to the beginning. In fact there are 67 blanks so the whole STRING is 86 characters long. Then I did the same for the base with 57 - 5 for the track. (That was the reason for the -5 between the wheels of the carriage.) It's not a perfect track — 2a is might be better.

100 TTS=CS+CS+CS+
CS+BS+STRINGS(67,128)

110 TBS=BS+BS+BS+BS+BS+STRINGS(57,61)

I don't want to print all that at once. Each string is 86 characters long and I can print only 32 at a time. I can cut these out using MID\$. This is a very useful command which lets you chop bits out of a STRING. It takes MID\$([string],start,length). For instance MID\$("ABCDE",2,3) cuts out the string "BCD". If you don't give the length all of the string to the right of the starting point is chopped out. I always print the last 32 characters but make a new string by chopping the last character off and putting it at the front. It's as though the string were printed on a strip of paper which is being pulled across the screen but the end has been glued to the beginning. This is a useful technique for those games to teach frogs the Green Cross Code. Why did the chicken cross the road? Because it was playing leap-frog!



100 PRINT@320,MIDS(TTS,61);
PRINT@323,MIDS(TBS,61);
100 TTS=MID\$TTS,66)+MIDS(TTS,1,66)
100 TBS=MID\$TBS,66)+MIDS(TBS,1,66)
200 GO TO 100

The train will go round and round. To make it more interesting I've put in a signal which is controlled by the space bar. If the signal is red when the train approaches it stops. The variable P flags between -1 and +1 whenever the space bar is pressed. Variables like this are often referred to by programmers as flags — how appropriate here!

Whether you travelled with me through the page or have taken a short cut to the end, here is the complete listing of the train with the signal

10 REM TRAIN DEC 83
10 CLEAR1000 C=112:CLS:F=1
20 P8=CHR\$(134+16)
30 FOR I=240 TO 345 STEP32
PRINT@I,P8:NEXT
40 BS=CHR\$(188):CS=CHR\$(131)
50 CS=CHR\$(123+C)+CHR\$(131+C)
60 CS=CS+CS+CS+CS+CHR\$(130+C)+"-"
70 BS="00000000000000000000000000000000"
80 ES=CHR\$(126)+STRINGS(5,128)
+CHR\$(133+C)
90 BS\$="00000000000000000000000000000000"
100 TTS=CS+CS+CS+CS+BS+
STRINGS(57,61)
110 TBS=BS+BS+BS+BS+BS+BS+
STRINGS(57,61)
120 KS=INKEY\$1 IF KS=CHR\$(32)THEN F=F+1
130 FL\$=BS IF F=1 THEN FL\$=CS
140 PRINT@217,FL\$,IF F=1 THEN RED
150 IF T=48 THEN 120 ELSE 160
160 PRINT@320,MIDS(TTS,61);
PRINT@323,MIDS(TBS,61);
170 T=T+1 IF T=96 THEN T=0
180 TTS=MID\$TTS,66)+MIDS(TTS,1,66)
180 TBS=MID\$TBS,66)+MIDS(TBS,1,66)
200 GO TO 100

HAPPY BIRTHDAY OFFER

Many thanks to all you Dragon owners who have bought the 'Happy Birthday Dragon' offer over the last few weeks.

It has come to our attention, however, that one or two of you have experienced problems with this cassette, one being with the game 'Happy'. In order to restart the game after the third frog has been run over, it is necessary to press the spacebar on the keyboard.

The other problem that has come to our attention is that if you try to load 'Santa Laverna' or 'Mason Red' without first clearing the memory of 'Happy', this can cause a number of errors to appear while attempting to run the programs. Therefore, if you want to run either 'Santa Laverna' or 'Mason Red', then switch off your Dragon to clear the memory, type in CLOAD 'Laverna' or CLOAD 'Mason'. This will allow the programs to load without any difficulty.

DRAGON 64: LOW-COST PROFESSIONAL COMPUTING

With the launch of the Dragon 64, together with the OS9 Operating System, we have our passport to the world of truly professional computing. The price of the computer itself is less than is normally paid for the software it supports — so Dragon Data has slashed the prices of several well-known professional packages for use on the new machine.

In appearance, the 64 looks just like the 32, except that it is grey in colour. All the familiar input/output ports are there, with the addition of one more 5 VD — this is the RS232 (serial) interface. When you turn on the power, you still find little difference between the 64 and the 32. In fact, the command ?MEM! gives the usual value of 32871. This is because a great deal of care has been taken to make sure that most software for the Dragon 32 will also run on the Dragon 64. So on power-up, we enter a "32K mode". To move to the 64K mode, we need to type ?EXEC. (In EXEC488212) A blue cursor reminds us that we are in 64K mode, and the command ?MEM! now gives 41341. (The commands CLEAR and PGCLEAR can, of course be used to increase this value.) The whole 64K of memory is now available to RAM — a re-assembled version of the BASIC interpreter resides in the top 16K, but this can be over-written if it is not required.

Apart from the obvious advantage of extra memory (16K if you are writing in BASIC) and 32K for machine code (and there are several other new features), perhaps the most important is the RS232 interface, which allows communication with serial devices such as printers and graph-plotters or with other computers. New commands of the form DLQAD and DLQADM make it possible to use this port as easily as the tape interface, but it is also possible to send and receive single characters using simple routines described in the "DRAGON64 Supplement" manual.

Another useful modification is the Keyboard Auto Repeat facility which makes all keys repeat when held down. This is available automatically in 64K mode, and can be incorporated into the 32K mode by executing a simple routine. The delay before the Repeat Facility is activated, and the speed of the repeat, are adjustable.

In the 64K mode, the response of the keyboard is greatly improved — touch typists in particular will find that they are able to type normally

without such problems as "PRINT" coming out as "PRINT!"

Searching tapes for files is made considerably easier, in both 32K and 64K modes, by the fact that SKTPF, CLOAD and CLQADM do not give IO ERRORS when the tape is started in the middle of a file. Instead, the beginning of the next file is found before any attempt is made to load or skip.

A minor enhancement is that the reverse slash "\\" is available directly from the keyboard, using the SHIFT and CLEAR keys together.

The DRAGON64's non-standard form of USR function operation is brought into line with normal Microsoft BASIC. With a 32, the USR function will default to USR89 unless an expanded form such as USR81 is used (when the function is called, not at the definition stage). This is "just right" in the 64, so that the usual form of the statement (e.g. X=USR10) can be used. Unfortunately the form USR81 is now illegal, and so is a minor incompatibility between 32 and 64 — easily remedied by deleting the unwanted "\\".

No "reverse BOOT" is available to return from 64K mode to 32K mode, but a cold-boot can be forced by POKEing a 29 into address 113 and pressing RESET. This results in a return to 32K mode, but also "NEWs" any BASIC program in memory. However, the contents of the lower 32K of RAM are preserved.

No cartridges can be used in the 64K mode, although they will work as usual in the 32K mode. This means in particular that the DRAGON64's Disk Operating System cannot be used in 64K mode. DRAGON64S can, however, be used to BOOT a more powerful Operating System — OS9.

Its ability to support OS9 takes DRAGON64 out of the realm of "mass home computing" into the world of serious business computers. OS9 is a UNIX type Operating System, which is both versatile and easy to learn and use. With it you can load a variety of languages — BASIC, OS9 (a very much more powerful BASIC with a large number of extra commands), PASCAL and "C" are already available. A 512 column by 24 row text screen is provided for your own programming, or for use with the professional packages, such as STYLORAPH (an advanced, but simple-to-use Word-processor), RMS (a complete Record Management System), and DYNACALC (a computerized spreadsheet). OS9 and the packages it supports will be reviewed more fully in future editions of DRAGONWORLD.

What has the DRAGON64 to offer the home user — without disk drives and sophisticated operating system? If your main interest is in playing COSMIC INVADERS or BERSERK, then the answer is, not much. But if you want to make a little more practical use of your computer, then the extra 16K available under BASIC control can be very handy. A typical area in which 32K seems to get eaten up all too quickly is the Database Retrieval System.

A Database is the computerized equivalent of a manual "card index". Each Database contains a



number of RECORDS (card) and every record has entries in each of a number of FIELDS. For example, an address/telephone list will contain a record for each person. The first field could be the surname, the second field the forenames, the third field the address, the fourth field the postcode and the fifth field the telephone number. A record may be retrieved by specifying any of the fields — normally we would probably specify the surname, to obtain the address and telephone number, but we could specify the telephone number to discover whose number it is!

The following BASIC program is a simple Database — for the sake of brevity it contains only a small number of entries, but it can be extended into a more advanced system.

```

1 REM DATABASE
2 REM A D MAYER, 1984
10 POLEAR1 CLEAR10000: B= 10500
  C= 10000
20 GMBS1,5)
30 DATACREATE DATABASE,
  LOAD DATABASE,400 RECORDS,
  DELETE RECORD1,FINO RECORD1,
  KILL DATABASE,SAVE DATABASE
40 SS01= "OL_ SS11= "AD105"
  NR2= 2,N11= 5
50 FORL= 1TO1 FORJ= 1TON1L
60 READSS1,L,NEXTJ,L
70 L= 2 GOSUB1200
80 ONX GOTO099,300
90 L= 1 GOSUB1200
100 ONX GOTO040,500,600,700,800
200 GOSUB1200
  INPUT"NUMBER OF FIELDS":F
210 INPUT"AVVERAGE FIELD LENGTH":AV
220 GOSUB1300: N= 0
230 FORJ= 1TOF
240 PRINT"NAME OF FIELD":J
  INPUT$1,J,NEXTJ
250 ONX1200
260 GOSUB1300
270 GOSUB1400: OPEN"1",J-1,NS
280 INPUT#1,F,AV,N GOSUB1300
290 FORJ= 1TOF-1 INPUT#1,J,$0
340 FORJ= 1TON-1
  INPUT#-1,ASJ,J,NEXTJ,L
350 CLOSE#-1 GOTO090
400 IFN= 0Z THEN170 ELSE N= N+1
  CLS,PRINT"RECORD NUMBER":N J= 0
410 PRINT$1,J
420 INPUT$1,IFX5= "THE N448
430 ASJ,J= 1,X3 J= J+1,IFJ<F
  THE N448 ELSE400
440 N= N-1,000,000
500 J= 0 GOSUB1200
510 GOSUB030,IFJ= N THEN90
520 GOSUB1500 PRINT
  "THIS RECORD": GOSUB1100

```

```

530 IFX5= <"Y" THEN90
540 N= N-1,J= J-1
550 IFJ= N THEN90
560 FORJ= 1TOJ-1 ASJ,J= ASJ,J+1
570 NEXTJ,J= J+1 GOTO060
580 IFJ= N THEN120 ELSE510
590 J= 0 GOSUB1500 FL= 0
610 GOSUB040,IFJ= N THEN90
620 GOSUB1400,00SUB1100,IFJ<N
  THEN120,L520
630 IFPS= ABSF= 1,J)THENRETURN
640 J= J+1,IFJ<N THEN90
650 IFFL= 12 THENPRINT"NOT FOUND"
  GOSUB1100
660 RETURN
700 PRINT"KILL": GOSUB1100
  IFX5= <"Y"THEN120 ELSE1100
800 GOSUB1400 OPEN"1",#-1,NS
810 PRINT#-1,F,AV,N
820 FORJ= 1TOF-1,PRINT#-1,FSJ,J
830 FORJ= 1TON-1,PRINT#-1,
  ASJ,J,NEXTJ,L
840 CLOSE#-1 GOTO090
1000 CLS FORJ= 1TON1L
  PRINTTAB11,1,00SUB1100,1,
  TAB10,1,00SUB1100,X= INSTR1,SS1,L,0X)
  IFX5= 0 THEN10 ELSE1100
1100 X$= INKEYS
  IFX5= " " THEN1100 ELSE1100
1200 INPUT"NAME OF DATABASE",
  NS,IFLEN(NS)>8 THENRETURN
1300 PRINT"100 LONG": GOTO1200
1300 IFN<0Z+2>C THENN2=
  INT(C/P)-2
1300 DIMSP#1,AS#-1,NZ#RETURN
1400 PRINT"PRESS SPACE WHEN READY"
1410 GOSUB1100 IFX5=<>
  THEN1420 ELSE1100
1500 FL= 1 CLS,FORJ= 1TOF-1
  PRINT$1,J= "ASJ,J"
1510 IFPEEK1361>256= PEEK
  (137)<1472 THEN1500
1520 GOSUB1100 CLS
1530 NEXTJ,J= J+1 RETURN
1600 INPUT"FIELD NUMBER":F
1610 INPUT"TARGET":T$ RETURN
1700 PRINT"DATABASE FULL":
  GOSUB1100 GOTO090

```

As it stands, this program will work with a DRAGON32 or a DRAGON64 in 32K mode.

Line 101 POLEAR1 to 1 graphics page (the minimum allowed without "POLEAR1" to "POLEAR2"), and then distributes the remaining memory between string and non-string categories, in an attempt to make best use of it. Lines 20-60 set up strings for the menu. Lines 70-80 and 90-100 display the menu and use subroutine 10000 to interpret the user's response — line 10100 uses INSTR on the appropriate \$0 to decipher the response.

DO YOU TAKE YOUR DRAGON SERIOUSLY???

MAKE YOUR DRAGON EARN ITS KEEP WITH ONE OF OUR APPLICATION PROGRAMS

JUNIPER BASIC WORD PROCESSOR

A menu driven BWP with unique full screen editor. Built functions include auto repeat, basic character set, tabulation auto repeat, upper/lower case, etc. Print functions include justify, no word break, up to 200 printed characters per line, multiple copies, double space, PROG SLASHED TO DR 80.

DATATEX 1040 COMPUTER CASSETTE RECORDER £29.95

TELEWRITER ADVANCED WORD PROCESSOR

A more advanced word processor with additional features such as term display with four line case block move, find and replace, etc. Memory 128 KB.

DRAGON MORSE TUTOR

Designed to teach Morse up to a Amateur Radio Morse Exam standard, and beyond. The programme has full levels and allows the user to select letter spacing, letter or symbol to record. Record range 8-32 wpm. A bargraph at 16 KB.

PERSONAL FINANCE MANAGER

A suite of 8 programs plus comprehensive user manual designed to aid the management of your domestic or small business accounts. The PFM is an easy to use menu driven guide to every general ledger which includes -

- ✓ user defined date formats for optimum flexibility -
- ✓ Typically 100 accounts
- ✓ On screen or printed reports
- ✓ User defined transaction code
- ✓ Performance reported budget

CRU 17.95

TELETUTOR EDUCATIONAL PACKAGE

An educational package from 80 modules suitable for ages 5 to adult. Includes spelling, math, word skills, science and estimating. CRU 80.

JOYSTICKS

Plastic (yellow mesh) £19.95
Soft (leather) (white) £21.95
£15.95
RED SLICKER-102 PISTOL,
GIMP JOYSTICK NOW
AVAILABLE ON DRAGON - £19.95 (switched controller)

DRAGON DUST COVER WITH LOGO £4.95

LANGUAGES: DRAGON FORTH £14.95 DRAG BASIC COMPILER £14.95 DRAG PASCAL COMPILER £14.95
CHESS: CYRUS CARTRIDGE £14.95 DRAG BASIC CASSETTE £9.95

PRINTERS ** NEW LOW PRICES **

DISK DRIVES

COLOUR MONITORS

STAR ST4000-80cps 8-1/2" wide paper, Proton feed, thermal dot matrix printer. Supplied value at £149.95 Thermal paper £5.50 per 100 £5.00
STAR SP4510 105cps 128 column, Proton/Inkjet feed, true descenders, sub/superscripts, italics, 24 buffer. Price reduced to only £29.95
SHARMA CT-CP480 96 column, Proton/Inkjet feed, true descenders, sub/superscripts, italics, etc. etc. Now available for only £29.95
JURO 8100 dot-matrix printer 110cps, up to 200dots/line, bold, shadow, underscores, and many more features. Supplied value at £29.95

Printer cable to connect any of above printers to your Dragon 32 £15.95
User Telewriter and other Dragon software with printer. £55.95 printer interface, plug into Centronics port on Dragon. £42.95 Dragon Data Double density anger disk drive 8 controller £20.95 Lockable disk storage box with pegs and lid. Holds up to 40 disks £17.95

OPTIONAL OFFICE UNITS

20th Feb 88 £10.10 Single sided double density disks £17.95

Fidelity CM14 Colour Monitor with Composite video, RGB and audio inputs £229.95 Colour CM14 to Dragon £9.95

PROGRAMMING If you have any serious application software for the Dragon 32 or 64, please send us a sample. Good rates for quality software.

GAMES — We can supply a comprehensive list of games including The King, Frogger, The Outback Range, Shuttle, Crazy Poles, Danger Range, Classics, Scramble, Dragon Hawk, Pong, Eight Ball, Space Fighter, Space Rangers, Air Traffic Control, City Assault, and Interceptor Force from Microtel (at £9.95) plus games from Dragon Data, CASIC Software, Flexsoft, Rainhamsoft, Games, Quark Data, Software Data.

PRICES All our prices are inclusive of VAT and carriage to UK
remitted. Transfer, semi-cheque, P.O. or Access No. Or phone
Your Address No.

OPTIONAL OFFERS 10% Discount on 3-4 cassettes
Send for our £110.00 price list of DragonOric Hardware
Software, and Accessories.

We reserve the right to amend prices without notice

JUNIPER COMPUTING

8 PEMBROKE GREEN, LIA, MELINSBURY, WILTSHIRE SN10 5PF. Tel. 041-41-261

SALE ON PHONE BILLS AND CALL US CHEAP RATE BETWEEN 8.00 p.m. and 7.00 a.m.

Addison-Wesley Publishers

CALLING ALL DRAGON OWNERS

TWO NEW BOOKS FROM
ADDISON-WESLEY
PUBLISHERS

As approved
by Dragon Data

INSIDE THE DRAGON

by Duncan Strand and
Tom Sommerville

For the serious hobbyist wishing to understand more of the technical details of his system, **INSIDE THE DRAGON** is a unique and fascinating journey through the interior of the Dragon. It considers the machine's internal workings and organisation, and brings together a wealth of information on both the software and hardware of the Dragon. A working knowledge of BASIC is assumed, making this book the obvious choice for those wishing to realise the full potential of their machine.

Duncan Strand is admirably qualified to provide this reader's book - he was responsible for the design of the built-in Dragon instructional routines for Dragon Data Ltd.

Topics covered include:

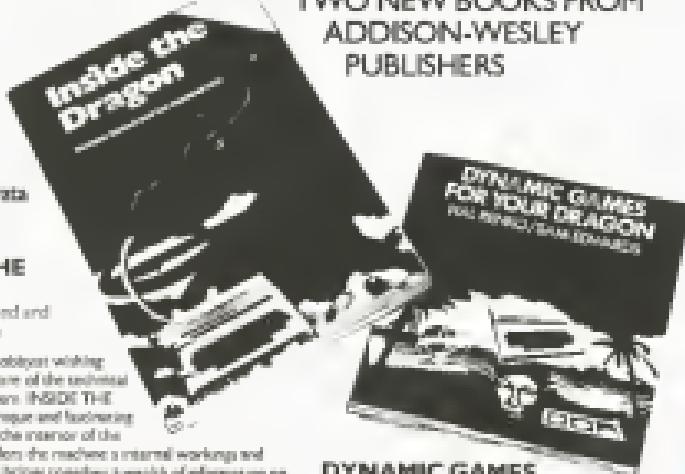
- the architecture of the M6809 - the chip at the heart of the machine
- input/output hardware
- graphics hardware
- the Dragon 64
- the disc operating system

Also detailed are the manufacturer's data sheets for the M6809 processor, the SN74LS183 multiplexer, the PC1640 video display generator and the PC16401 interface adapter.

380pp

201 14023 5

£7.95



DYNAMIC GAMES FOR YOUR DRAGON

by Hal Renshaw and Sam Edwards

Create for yourself hours of excitement and entertainment with this original and imaginative collection of games for your Dragon. Choose from adventure games, games of strategy and chance, puzzles, quizzes and competitions, and mind-savers. So tone up the muscles of your mind and test your reflexes and mental prowess on this extraordinary obstacle-course.

Games Included

- Zombies in the Swamp
- Kentucky Derby
- Galactic Monsters
- Carrie at Work
- Keyboard Memory
- Shakespeare Shuffle
- Las Vegas Go-go
- Adventure of the Century, II
- Parrot
- The Swedish Pop Song

140pp

201 14023 8

£9.95

Addison-Wesley Publishers

Should you have difficulty in collecting from your local bookseller, please fill in the coupon and add £1.50 p&p per book.

14023 5 Strand/Inside the Dragon £7.95
 14023 8 Renshaw/Dynamic Games for your Dragon £9.95

I enclose my cheque for £..... D6.
Please charge my Access/Barclaycard/Visa/American Express/Diners
Club Account No. _____

--	--	--	--	--	--

Signed Date

Name

Address

Dept C
Addison-Wesley Publishers Ltd.
52 Bedford Square
London WC1B 3DE
01 580 1101

The first option one to Create a (new) Database or to Load an existing one. In either case, a name must be specified for the Database. If a Database is loaded, the program proceeds directly to the second menu. If the Create option is used, an average field length must be estimated, names must be given for all fields and then the data are entered. "End of data" is indicated by pressing <CTRL>Pc on its own.

The second menu allows the Database to be Saved (on tape) — the file will be given the same name as the Database. Records may be Added — effectively a return to the "Create" mode. The Database may be interrogated, using the Find option — a field number is specified, then the particular record in that field (in our address) Telephone Database, field number 1, followed by the surname of the particular person, for example A record may be Deleted — it must first be "found" using a Record-type sequence. Finally the Database may be "Killed", usually only after Saving to allow a new Database to be Loaded or Created.

To convert to 64K mode, only line 18 need be altered. The maximum number of records allowed is calculated automatically by subroutine 13000 and is controlled by the parameter values in line 18. In 32K mode, after the program is loaded, there are about 27000 bytes available for data. Each record requires 6 bytes of non-string memory (essentially a pointer to indicate where the record can be found in RAM) and the relevant number of bytes (equal to its length) in string memory. If a reasonably accurate "average field length" is given, the program will calculate the number of records allowed and this number should nearly fill both memory areas. The numbers specified assume an average string length of about 12 for maximum efficiency. B is the number of bytes of string memory (reduced by 600 bytes to allow for any slight underestimate of the average), and C is the total number of records the allowed — this is limited by the non-string memory available. So $B + C$ must not exceed the available memory (27000 bytes). In this case, $B + C = 26820$, which allows 600 bytes to spare in both types of memory. In 64K mode, however, the available memory increases to 42000 bytes. So we need B and C to satisfy $B + C = 21000$ to allow the same margin of error. B = 29600, C = 2400 are suitable values, so line 18 should be replaced by

18 POLEAR1.CLEAR\$99999 B = 29600 C = 2400

To obtain the optimum values of B and C for a particular average length, AV , use the formulae

$$C = 26620/(AV + 5)$$

$$B = 26620 - 5 \cdot C$$

(For 64K mode, replace 26600 by 41000.)

\$7F00 to \$7FFF into A and thence to the printer

The assembly of the program was achieved using ALCD ROM situated in RAM from 23000 onwards (see elsewhere in this issue for relevant detail) and the listings were produced using DREAM and DREAMBUG.

If you wish to implement the program from BASIC then you must construct a BASIC program similar to the one below (which gives the equivalent version for the Sekkosha). To do this change line 18 to FOR I = \$H70000 TO \$H7D80 and then supply DATA statements using the listing of bytes from the assembly listing (8 bytes, 68, 18, 80, 60, 0F, 30). Once you may record both programs using CSAMEM or the equivalent DOS command.

Finally note that to use the programs, you must first make a CLEAR command such as CLEAR\$10000 BH70000, and then invoke the program with EXECBH7000.

5. ROM MACHINE CODE SCREEN DUMP FOR SEIKOSHA GPIB8A

```

19 FOR I = $H7D80 TO $H7D89
20 READ A$: POKE I,VAL(A$)+$01:NEXTI
21 DATA 9E,5A,1F,19,FD,7E,24,1F,1,10
22 DATA 9E,7F,23,18,19,CD,9C,01,FD,7E
23 DATA 20,9E,99,1C,6E,1F,13,7F,7E,23
24 DATA 7C,7E,22,11,31,87,24,13,45,24
25 DATA 20,3,FB,7E,23,79,7E,22,39,09
26 DATA 20,11,83,7E,29,26,EB,8,87,49,10
27 DATA 8C,89,9,27,8,81,1,27,3,44
199 DATA 20,1C,20,1,29,01,10,8E,7F,0
110 DATA A6,A2,81,88,26,8,18,8C,80,0
129 DATA 29,4,29,18,8E,77,9,25,8
139 DATA B0,88,7,4A,8A,8D,8D,8J,10,8C
140 DATA 60,0,28,FB,23,8,88,0,8D,8D
158 DATA F,FC,7E,24,C,3,8,8B,18,89,87
168 DATA 25,8D,8E,FB,8D,8D,7
```



Features include:

- 255 MAXIMUM NO. OF ROWS
- 255 MAXIMUM NO. OF COLUMNS
- VERY EASY TO USE
- INDIVIDUAL CELL FORMULAE
- COPY BLOCKS OF CELLS
- FULL CELL-EDIT CAPABILITY
- COMPATIBLE WITH ALL PRINTERS
- EASY 100 COLUMN PAGE WORK
- GRAPH FORMAT FOR BAR CHARTS
- COMPREHENSIVE MANUAL INCLUDED
- INTERNATIONAL USER GROUP
- SORT ROUTINE IN ASCENDING OR DESCENDING ORDER

- Single character commands
- Help display
- Enter and/or format up to 255 characters
- Report and witness
- Available memory status displayed
- Page display option for test entries
- Graphics automatic colour selection
- Insert/Delete. Move entries between columns
- Any cell can be set to fit a certain column width with automatic line adjustment
- 40 machine language for maximum speed
- 2000 data storage spaces database in 2000 bytes
- Data entry formats
- Arithmetic operations (+, -, ., ., /, =)
- Relational operations =, <, >, <=, >=, <>
- Logical operations AND, OR, NOT
- Conditional formulae IF, THEN, ELSE
- Test Functions ISBLANK, ISNOT, ISBLANK
- Log. Functions LOG, EXP, SQRT, SIN
- Rec. Functions INT, TRUNC, RND
- Range Functions SUM, AVERAGE, COUNT, MIN, MAX, LOOKUP
- New logic functions
- User definable constant table
- User definable printer set-up commands
- Individual column width redefined to 255
- Adjustable line height to meet store line without memory memory
- Print columns in report
- Multiple print font selectable cell by cell
- Display/Print formats cell by cell, user definable
- E format comma grouping, plus or minus signs
- Scientific notation fixed point and large formats
- Left or Right cell content justification
- All cell formats
- Informal stored with worksheet on disk
- Standard disk tape film in compact memory form
- Screen line directions
- Colour ASCII for word processor input compatibility
- Memory resident code - no required disk code

ELITE

C
A
L
C

NOW... The worksheet calculator program you've been waiting for is waiting to work for you. ELITE*CALC is a powerful, full featured spreadsheet for the Dragon 32/64 and Tandy Color Computer. Answer 'what if' questions, prepare reports and cash flow projections, maintain records and perform other tasks which, until now, required sophisticated business computers. ELITE*CALC is a serious tool for those who want to do more than play games on their micro.

Available from stock on cartridge for the Dragon 32 or 64 or Tandy Color Computer... please specify *Sheety* available on disc for the Dragon Data or Commander/Premier or Radio Shack systems. Also available in the *B*-format instead of the *E*-format if requested.

THE BEST
FOR ONLY

£45

Elite*Calc has had excellent reviews in the American press and an enthusiastic reception at the recent International Rainbowfest.

"Elite*Calc is a great spreadsheet program."

Stuart Hawkinson, Rainbow

"Truly one of the best programs I have seen."

John Steiner, Micro

"Very powerful program... essential to every serious user."

Mike Jarvis, M & J Software

ENQUIRIES INVITED FROM RETAILERS AND
DISTRIBUTORS (HOME AND OVERSEAS)

MICROCARE
1 OAKWOOD ROAD,
RODE HEATH,
STOKE-ON-TRENT
ST4 (09363) 5695

DRAGON 32/64

TANDY COLOR COMPUTER

The family favourites for your Dragon 32



NEW RELEASES

MYSTERY OF THE JAVA STAR: Four part (over 100K) educational adventure for the older child. 3 levels of difficulty. Packed with challenges, puzzles and information. £7.95

SHAPERS: Comprehensive sound effect utility. Create, store and modify machine code sound effects, for use in your own programs. Comes with a store of 160 effects and a comprehensive manual. £9.95

NORTH SEA OIL: Simulation game. Run your own oil rig and see how fast you can make 100 million dollars! £5.75

INTERACTIVE MATHS: Four 32K software for the whole family. For last math lessons. £7.95

UNISITE: Highly popular strategy game. £5.95

PLAY IT BIG: Action packed game. 4 levels. £5.95

SHOOTERS: Fast action shooting game. 12 players. £5.75

FUN TO LEARN: Maths based education for 6-11 year olds. £5.95

WIN STYL MATHS: Maths based education for 8-11 year olds. £5.95

FAMILY PROGRAMS: Comprehensive software

PLAY & LEARN: Action based education for all ages. £5.95

LEAP & LEARN: Educational competition for older children. £5.95

QUIZ PAGE: Test your brains. 100 rounds. £5.95

INFANT PACK: Pre school education. Counting and alphabet. £5.95

JUNIOR PACK: Primary school education. £5.95

DEALERS! Please call now
or 01-516 4871 for our
comprehensive catalogue



Available from selected branches of Boots and all good stockists, or by telephoning Access/Bardseyard
orders to 01-516 4871 or by sending cheque/PO to: 109 EDEN ROAD, ILFORD, ESSEX, IG1 2UQ



THIS MONTH ...

SOFTWARE REVIEWED

VENTURE FORTH

The unusual computer language FORTH is now available from DRAGON DATA on a cassette for use on DRAGON 32 or 64. A FORTH program will normally run faster and use less memory than an equivalent BASIC program — although slower than machine code. It is exceptionally efficient for a high-level language.

At first sight it is an odd-looking language. It employs reverse Polish notation (RPN), which is the same as that used by some Hewlett-Packard calculators. Thus the BASIC expression $5+3^2$ becomes $5 3 2 ^ +$ in FORTH.

The main reason for using this unusual notation is that FORTH uses stacks. This in itself is not unusual, since all languages use stacks internally, but FORTH interacts with the programmer via stacks. In fact, virtually nothing can be done in FORTH without using stacks. When a number is entered, it is PUSHED on to the stack. When an operator is entered, two numbers are PULLED from the stack, the operation is carried out, and the result is PUSHED on to the stack.

FORTH consists of a standard vocabulary of "words". A word is roughly equivalent to a subroutine in BASIC. DRAGON FORTH is particularly powerful, since it includes a number of extensions to the standard FORTH vocabulary. The real strength of DRAGON FORTH, however, lies in its direct access to most of the DRAGON's BASIC commands, including the high-resolution graphics commands.

The basic principle of FORTH programming is the use of the existing vocabulary to define your own new words, which in turn may be used to define more words. New words are defined using the colon. Thus the FORTH statement

POWER: DUP * DUP .

creates a "subroutine", which takes the number at the top of the stack, finds its fourth power, and prints out the result. The operator "DUP" duplicates the top number on the stack, so that when the "*" operator is applied, the result is the square of the original number. This process is repeated to create the square of the square, or the fourth power. Then the "*" operator prints out the result. The semi-colon indicates the end of the definition of the word "POWER".

Once POWER has been defined in this way, any subsequent statement of the form **3 POWER** will push the number (in this case 3) on to the

stack, then calculate its fourth power (81) and print the result.

DRAGON FORTH allows you to access BASIC via commands of the form **B1 BASIC** statements. One-line BASIC programs can be included, but commands which need line numbers (such as GO TO) cannot. Thus the command

B1 PRINT "THIS IS A STATEMENT"

will print "THIS IS A STATEMENT", and the command

B1 PMODE3 SCREEN1.0 PCLS CIRCLE180.080.70.4 PAINT#8.80.241

will enter graphics mode 3, display the screen, draw a red circle and paint it yellow. It will also leave you in graphics mode, making it difficult to see what is happening on the text screen! A simple statement like **B1 PRINT** can be keyed in, and will return you to text mode, but it must be done "blind". It is useful to note that the "1" form of "PRINT" also works.

DRAGON FORTH also includes a comprehensive line editor, which allows you to write FORTH programs in "pages" of text. These pages can be saved on tape, by executing BASIC and using CSANTIM, so that they can be reloaded later. They can also be compiled into the FORTH dictionary using the FORTH command LOAD, to be executed later.

To the programmer who has never used anything but BASIC, FORTH is something of a challenge. It doesn't use any of the "safety checks" that DRAGON's BASIC Interpreter performs, and this can make debugging difficult. If you are keen to speed up your graphics animations, and don't want to go all the way down to the level of machine code, the challenge is worth accepting.

The first step is to get to grips with the stack system and the reverse Polish logic. A program that simulates these two aspects at least as far as numerical operations are concerned and displays the stack as it is manipulated is an invaluable aid to those about to venture forth.

The program below is designed to do just that. It uses a stack of size 100 and allows you to enter integers and perform standard FORTH integer arithmetic, using the operators

- +** adds together the top two numbers on the stack
- subtracts the top number from the second number
- *** multiplies together the top two numbers
- /** divides the second number by the first, ignoring any remainder
- MOD** performs as %, but leaves both result and remainder on the stack
- DUP** duplicates the top number on the stack
- DROP** deletes the top number
- ROT** "rotates" the top 3 numbers, so 1 2 3 becomes 3 1 2
- SWAP** swaps the two top numbers
- PRINT** prints the top number.

The first five operators pull the numbers from the stack, operate on them and push the results on to the stack. The output operator " ." puts the number prints it, and does not replace it.

Type in the program and RUN. The stack is displayed in the middle of the screen and your FORTH instructions along the top of the screen. Key in numbers as usual, and use the keys 0 for DUP, 5 for SWAP, R for NOT, 4- for -, and / for their FORTH counterparts. In addition use the downward arrow for DROP and % for /MOD.

Although a single key is used for each instruction (allowing easy branching through an extensive use of INSTR) the equivalent FORTH commands are displayed. Unlike the version of FORTH produced by DRAGON DATA, the period . produces an immediate display. When your list of FORTH words is long (up to two lines of 60) you can clear the top of screen by pressing ENTIR which clears the list of previous commands but does not alter the stack.

In FORTH you have to be careful to leave a space between each instruction. Our BASIC program uses the space to implement the operation and an error message is generated if you do not follow this cardinal rule. If you make a mistake in keying in the last key pressed may be retrieved by use of the right arrow.

As a first attempt try the following sequence of commands (remember these spaces!)

102 5 + ." calculates 23! (32!+5)
103 5 % (standing for 23%5 SWAP /MOD which would give you 4 and 3 on the top of stack as 5 divides into 23 4 times with remainder 3)

It is possible to extend the program to allow other FORTH primitives but hardly worth it. The program is designed to introduce you gently to the essentials syntax of the language and no more. Perhaps such a fanatic might tempt you to your local stockist to purchase the real thing.

6 REM TRY FORTH - A. M. SYKES

DEC 1983

```
10 CLEAR1000:04000101,X00121.
CLS:GOSUB210
20 AB="0123456789":CHR$99:BS="""
30 BS=CHR$1321:+" "+CHR$1471
:CHR$1321:+"DSR":CHR$1120:CHR$131
40 FOR I=1 TO$X$01:MT$0$BS,11:NEXTI
50 FOR I=7 TO 11 READ X$01:NEXTI
60 DATA,MOD,DUP,SWAP,ROT,DROP
70 CCS="""
80 KS=INKEY$:IF KS="" THEN B0
90 A=INSTR11,BS,CCS
  B=INSTR11,BS,KS:IF A=0 AND
  B=0 THEN B0
100 IF B=13 THEN CCS=":":CCS="""
  PRINT$1,STRING$16,321:GOTO82
110 IF LEN(CC$)>0 AND A=11
  THEN CC$=BS:CCS=005:GOSUB200:
  GOTO80
120 DS=CC$:DOS=CC$:IF B>0
  THEN CC$=CC$+X$01:ELSE CC$=CC$+X$00
130 GOSUB200:CCS=CC$+KS:L=LEN(CC$)
140 IF L<2 THEN B0
```

160 BC=INSTR11,BS,0051

```
160 GOSUB200 IF B=1 AND BC>1
  THEN ON BC-1 GOSUB 260,270,280,290
  ,210,220,330,340,350,360 GOTO70
170 IF B=1 AND BC<0 THEN X$00
  -VALID$0:GOSUB240:GOSUB250
  GOTO70
180 IF B=0 AND BC<0 THEN B0
190 GOSUB200:GOTO80
200 IF B=1 AND BC=1 THEN B0
210 FOR I=1 TO B PRINT$1,BS+I-32,":"
  ,":",NEXTI:PRINT$1,BS+I-32,X$00,":"
  ,":",RETURN
220 PRINT$1,1,BS:RETURN
230 FOR I=0 TO 12 STEP -1
  X$00= X$01:NEXTI:GOSUB 250:RETURN
250 FOR I=1 TO 10 PRINT$1,11$+I-32,
  ,":",NEXTI:RETURN
260 X$11=X$01+X$01:GOSUB 370:
  GOSUB 250:RETURN
270 X$11=X$02-X$01:GOSUB 370:
  GOSUB 250:RETURN
280 X$11=X$01+X$02:GOSUB370:
  GOSUB 250:RETURN
290 PRINT$1,BS,"display ",X$11
  GOSUB290:RETURN
300 PRINT$1,1,BS,"error":
  IF I=1 TO 500 NEXT .CCS=005 CCS=005
  GOSUB230:RETURN
310 X$11=FD$X$12|X$010:GOSUB250:RETURN
320 W=X$01:X$01=FD$002|X$11
  X$01=002-W:X$01:GOSUB250:RETURN
330 FOR I=9 TO 1 STEP -1
  X$01= X$02:NEXTI:GOSUB250:RETURN
340 W=X$01:X$01-X$01:X$01=W
  GOSUB290:RETURN
350 W=30:30:X$01:X$01:X$01-X$01
  X$01=W:GOSUB250:RETURN
360 FOR I=1 TO 9 X$01=30+I
  NEXTI:GOSUB250:RETURN
370 FOR I=2 TO 9 X$01=X$01+1
  NEXTI:GOSUB290:RETURN
```

ALLDREAM

As mentioned in STOP PRESS 5 the cassette based assembler program DREAM is complemented with DREAMBUG, a program designed to provide the user with the tools to de-assemble machine code and de-bug programs. Together they form ALLDREAM, but if bought separately DREAMBUG comes complete with a program to combine both together ready for you to record a copy of the whole package.

Together they form a comprehensive toolkit for the machine code user. In particular the printer options in DREAMBUG make it extremely easy to obtain listings of assembler mnemonics from machine code programs in RAM or ROM.

ALLDREAM when loaded resides in RAM from hex \$0000 to the end (\$FFFF), which conflicts with the practice in this magazine of constructing machine code programs in RAM from say hex 7D800 onwards. Further, DREAM (under default instructions) assembles programs from \$000001 onwards. This means that it would be necessary



then to protect 2000000 onwards from BASIC by use of CLEAR and this would be unnecessarily wasteful of space, particularly for programs such as the INDEX program discussed in the last issue.

We prefer therefore when using ALLDREAM to load it into RAM further down leaving hex 7D800 onwards free for machine code. This may be achieved by using an OFFSET. To calculate the (negative) offset required let's assume that we wish to install ALLDREAM from 2000000 (instead of 247680) onwards. The necessary offset will be $2000000 - 247680 = 1552320$ and the command for cassette loading is

LOAD"ALLDREAM",1552320

(From disk the equivalent command would be **LOAD"ALLDREAM,BIN",2000000,1**) To assemble a program from hex 7D800 onwards simply insert the two lines **ORG \$7D800** and **PUT \$7D800** at the beginning of your 16bit file (see for example the issue 8 machine code corner).

Of course you have to adjust all the addresses supplied with ALLDREAM accordingly so that for example with our suggested position, EXEC 2000000 takes you to DREAM whilst EXEC 27466 takes you to DREAMBUG.

DRAGON DATA SOFTWARE REVIEWS

The month 'Shant' and 'Viking' have been reviewed by local schoolchildren. Descriptions of the software titles are given below, follow the scores and comments of the children.

Title: SHANT

Price £7.95

The aim of the game is to move across the screen from one side to the other avoiding the eight elevators moving up and down. You begin the game at the base of the screen and each time you successfully cross it, a special elevator will lift you up a level. After crossing the screen safely in every level, you will eventually arrive at the top, when you will be presented with a new, faster moving screen. Your reactions will have to get quicker and quicker for you to reach the top of the shant.

Marks out of 10

User	User
A	B
4	4
8	9
7	7
6	6
7	8
33	36

Comments

An enjoyable game with a lot of lasting appeal.
Good value for money

Title: VIKING

Price £7.95

In Viking you are given the chance to work your way up the ranks to become King or Queen. But first you must make decisions regarding what is to be bought and sold, what taxes to impose for the coming year, what food your people will need - all your decisions can have far-reaching effects. For example, taxation will affect pricing, food will affect the population. A disaster such as a terrible plague, raid or revolution may lead your kingdom but if you have the power to divert them or prevent them altogether, if you possess your wits, strength and ability you will rise to be the Viking Monarch.

Marks out of 10

User	User	User
A	B	C
3	4	4
8	8	8
9	9	9
8	8	9
36	36	38

Comments

A very interesting and enjoyable game. A lot of skill required.

(continued from page 8)

'Basic Tutorial', and its advanced complement 'Advanced Tutorial', and ready later this month are, 'Computer Science' and 'O Level Physics'. Under production and on the drawing board are, 'Ogigraphy', 'Biology', 'Astronomy', 'French', and 'German'.

Superspy is a brilliant combination of every popular video game skill in a unique and thrilling teach-yourself history format. This fun way of acquiring knowledge of British history 1066-45 has been cleverly devised by a top history teacher for 11 year olds and upwards.

The full list of software now available from Dragon Data at £14.95 is

Basic Tutorial	(2 cassettes)
Advanced Tutorial	(2 cassettes)
Maths 1 14-Year olds)	(2 cassettes)
Maths 2 10-7 Year olds)	(2 cassettes)
Maths 'O' Level	(2 sets of 2 cassettes)
Superspy	(2 cassettes)
Early Reading	(2 cassettes)
Computer Science	(2 cassettes)
'O' Level Physics	(2 cassettes)

Please see order form on separate sheet

PREVIEW OF 'INSIDE THE DRAGON'

One of the major criticisms levelled at the Dragon 32 and 64 was the lack of technical information available for them. 'Inside the Dragon' written by Duncan Simcock and Ian Sommerville and published by Addison Wesley has been written specifically to fill that gap.

Inside the Dragon is 390 pages long and consists of nine chapters of text, a suggested further reading list, eight appendices and a complete index. The first chapter introduces basic computer principles using, obviously, the Dragon as a specific example. The second chapter explores the architecture of the M6809 microprocessor covering such topics as the register set, addressing modes and memory mapped input/output. Chapter Three contains a detailed discussion of the M6809's instruction set. The next chapter introduces assembly language explaining the facilities that a typical assembler in this case Dragon Data's own Editor/Assembler package provides and how it is used to create machine code programs. The fifth chapter then goes on to explain how the various BASIC statements can be coded in assembly language and illustrates how good programming techniques can be used to simplify the writing of assembly language programs. This chapter concludes with a complete assembly language source program of a simple monitor. The sixth and final chapter devoted to assembly language programming includes the use of substrings with examples of character string manipulation. This chapter is also

used to explain advanced programming techniques in assembly language including parameter passing using the stack, recursive subroutines, position independent code and combining assembly language with BASIC.

The seventh chapter is the first to explore the Dragon's potential in depth and describes the machine's graphic capabilities and how they can be manipulated by the assembly language programmer. Among the topics discussed are graphics display hardware, integrating BASIC and assembly code graphics display modes, graphics utilities and designing and implementing graphics programs. This chapter concludes with a listing of a complete animated graphics program. Chapter Eight is devoted to a description of I/O programming techniques and includes full details of the Dragon I/O hardware and how it may be used. The ninth chapter concludes the book with hints and tips which include details of the power up and reset actions, how BASIC programs are stored, how BASIC represents strings and numbers, how parameters can be passed from BASIC to machine code programs and vice versa, how to extend the Dragon BASIC with new reserved words and facilities and finishes with a complete list of BASIC system variables.

The final 100 pages of the book consist of the appendices which contain the data sheets of the 4 major chips in the Dragon, details specific to the Dragon 64 and Disk Operating System and miscellaneous information such as BASIC token values and I/O jump tables.

Dragon Dungeon

GAMES PROGRAMS

Crystal Castle, Adventure, puzzle and memory adventure. £19.95
Temple of Zorin, 3D, Exploring adventure. £19.95
Dragon Castle, Real-time strategy. £19.95
Adventure Games, Introductory adventure for children. £14.95
Bugged!, Tactical arcade game with insect themed alien life forms. £19.95 (4 floppy disks)

EDUCATIONAL PROGRAMS

Maths Test, Basic, go where no mathematician has gone before! £19.95
Cancerous Cells, Maths Test, Maths is a disease! £19.95
Dragon the Brain's Reading Room, Reading for early readers who can read. £19.95
Advanced Algebra, When the world needs them these tools of experts and geniuses to be employed. £19.95
Maths Games, Maths just got easier for 10-14 year olds, plenty of visual pleasure. £19.95

SERIOUS STUFF

Cost-Plan Money Management for home and small business. £19.95
Calculator, Personal Math, The learning program that counts and counts to success. £19.95
The Manager Book of the Home or Office, Organisational skills. £19.95

NEW RELEASES

Alaric and the Dark Powers, by Mike McCormack £19.95

Dragon in the Dark, Dragon's first adventure was released in 1986 and here's the second companion, Alaric and the Dark Powers, £19.95 (4 floppy disks).

You should still take time to introduce Dragon in the Dark because the majority of you will consider this to be Dragon's first adventure over eight years in the house at the end of the year.

From early spring, Alaric's companion in the Dragon-Doc Dan Club for the first season is Alaric and the Dark Powers.

Dragon Digits, by W. J. Hurford £9.95

Aktron's collection of 100 puzzles from the International Science Maths Museum.

Fun with Numbers, by W. J. Hurford and L. J. Lomax, 100 mathematical hours of challenge for players from 5 to 12 years.

First Numbers, by Margaret Readman £9.95

Fun musical, graphics, programs for 4 to 6 year old children to help them learn.

Locating Dragon, programs will teach young players how to search Dragon's dungeons, a lesson and a test.

Dragon's Eye, from the game company of the same name, £19.95 (4 floppy disks).

DRAGON OWNERS CLUB

The Dragon Dungeon Club is a club run by Dragon 32 owners for Dragon 32 owners. Members receive the monthly newsletter, Dragon's Teeth, which includes:

- Book reviews (Dragon Standard) and news of forthcoming publications
- Hardware reviews (printers, disk drives, game consoles and add ons)
- Software reviews of games utilities, tools, news and educational programs
- Letters from our 1000 members and general discussions
- Tips and advice (including price tip of the Month)
- Machine Code Corner
- A review of the US 3200 game
- Competitions and Special Offers - 10% off all Dragon purchases
- Members' advertisements (free)
- Free listing without the members without payment)

The Dragon Dungeon Club also offers an advice service. If we can't help you with your Dragon problems, don't worry, we'll put you in touch with someone who can.

Membership of the Club costs

Annual Subscription £6.00 Overseas £9.95 £14.95

FOR DEALER AND WHOLESALER INQUIRIES PHONE 0333 339363

AVAILABLE AT 2000 SOFTWARE SHOPS OR BY MAIL ORDER

DRAGON DUNGEON, MILTON KEYNES, MK7 6JY, ADDRESSES, TELEPHONE 01908 452295, FAX NUMBER 01908 452296, FAX HOURS 9.00am to 5.30pm

